# REPORT DOCUMENTATION PAGE

*Form Approved*
**OMB No. 0704-0188**

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

**1. REPORT DATE** *(DD-MM-YYYY)*
MAY 2013

**2. REPORT TYPE**
CONFERENCE PAPER (Postprint)

**3. DATES COVERED** *(From - To)*
JAN 2012 – APR 2013

**4. TITLE AND SUBTITLE**
RANK DEFICIENT DECODING OF LINEAR NETWORK CODING

**5a. CONTRACT NUMBER**
IN-HOUSE

**5b. GRANT NUMBER**
N/A

**5c. PROGRAM ELEMENT NUMBER**
62788F

**6. AUTHOR(S)**
Zhiyuan Yan (Leighigh University), Hongmei Xie (Leighigh University), Bruce Suter (AFRL)

**5d. PROJECT NUMBER**
T2AW

**5e. TASK NUMBER**
IN

**5f. WORK UNIT NUMBER**
HO

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

**PRIME**
Air Force Research Laboratory/
Information Directorate
Rome Research Site/RITB
525 Brooks Road
Rome NY 13441-4505

**SUB**
Leighigh University
3941 Southwood Dr
Easton, PA 18045

**8. PERFORMING ORGANIZATION REPORT NUMBER**
N/A

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Research Laboratory/Information Directorate
Rome Research Site/RITB
525 Brooks Road
Rome NY 13441-4505

**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFRL/RI

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER**
AFRL-RI-RS-TP-2013-039

**12. DISTRIBUTION AVAILABILITY STATEMENT**
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA Case Number: 88ABW-2012-6128
DATE CLEARED: 26 NOV 2012

**13. SUPPLEMENTARY NOTES**
© 2012 IEEE. Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP, Vancouver, Canada; 26-31, May 2013. One or more of the authors is a U.S. Government employee working within the scope of their Government job; therefore, the U.S. Government is joint owner of the work and has the right to copy, distribute, and use the work. All other rights are reserved by the copyright owner.

**14. ABSTRACT**
Network coding is a new paradigm of communication networks that promises advantages in throughput, robustness, and complexity. Since the fundamental premise of linear network coding is that transmitted data packets are subject to linear combinations, for all network coding schemes so far, a full rank of received packets is required to invert the linear mapping so as to recover the transmitted data packets. This requirement unfortunately results in a key drawback of network coding: either all the packets (or bits) in a session are recovered simultaneously or none can be recovered. Aiming to overcome this all-or-nothing property, leading to long delays and low throughputs. This work proposes a variety of rank deficient decoders of linear network coding. To this end, two classes of rank deficient decoders are proposed. Both classes of decoders recover data from fewer received packets and hence achieve higher throughputs and shorter delays than the full rank decoder.

**15. SUBJECT TERMS**
Network Coding; Compressive Sensing; Rank Deficient Decoders

| 16. SECURITY CLASSIFICATION OF: a. REPORT | b. ABSTRACT | c. THIS PAGE | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| U | U | U | UU | 6 | MICHAEL LITTLE |
| | | | | | **19b. TELEPHONE NUMBER** (*Include area code*) N/A |

# Rank Deficient Decoding of Linear Network Coding

Zhiyuan Yan, Hongmei Xie, and Bruce W. Suter

***Abstract*—Network coding is a new paradigm of communication networks that promises advantages in throughput, robustness, and complexity. Since the fundamental premise of linear network coding is that transmitted data packets are subject to linear combinations, for all network coding schemes so far, a full rank of received packets is required to invert the linear mapping so as to recover the transmitted data packets. This requirement unfortunately results in a key drawback of network coding: either all the packets (or bits) in a session are recovered simultaneously or none can be recovered. Aiming to overcome this all-or-nothing property, which leads to long delays and low throughputs, in this work we propose a variety of rank deficient decoders of linear network coding. To this end, we first reformulate the decoding problem of linear network coding as a collection of underdetermined systems. This reformulation reveals the connection among the decoding problems of network coding and error control coding, and enables rank deficient decoding. We then propose two classes of rank deficient decoders. The first class of decoders take advantage of the sparsity inherent in data and produce the data vectors with the smallest Hamming weight, and hence they are called Hamming norm decoders. Since these decoders have high complexities, we propose a class of decoders based on linear programming, referred to as linear programming decoders. Considering linear programming relaxation of the Hamming norm decoders and solving them by using standard linear programming procedures, the linear programming decoders have polynomial complexities and are much more affordable. Both classes of decoders recover data from fewer received packets and hence achieve higher throughputs and shorter delays than the full rank decoder.**

## I. INTRODUCTION

Communication networks (CNs) are ubiquitous in our everyday life as well as our national infrastructure. Network coding [1] has the potential to fundamentally transform current and future CNs due to its promise of significant throughput gains. Furthermore, network coding has other advantages such as robustness and can be implemented in a distributed manner with random linear network coding (RLNC) [2]. Hence, network coding is already used or considered for a wide variety of wired and wireless networks.

Although network coding does not suffer from the coupon-collector problem, one significant drawback of network coding is its all-or-nothing property in more than one sense. First, a full rank of received packets at the receiver nodes of a multicast (or a unicast) is needed before decoding can start, leading to long delays and low throughputs, especially when the number of packets of a session is large. This is particularly undesirable for military or civil applications with stringent delay requirements. Second, all the bits in any packet are equal in the sense that they are recovered simultaneously.

Aiming to solve this problem, we propose rank deficient decoding for linear network coding, which can start even when the rank of the received packets is smaller than the threshold. By reformulating the decoding problem of network coding in a different fashion, the decoding problem reduces to a collection of syndrome decoding problems, where the code is defined by the global kernel matrix and its minimum distance is upper-bounded by the rank of the received packets. Solving these syndrome decoding problems, rank deficient decoding leads to smaller delays and higher throughput, at the expense of possible decoding errors. Specifically, we propose two classes of rank deficient decoders. The first class of decoders take advantage of the sparsity inherent in data and produce the data vectors with the smallest Hamming weight, and hence they are called Hamming norm decoders. Since these decoders have high complexities for large size systems, we propose a class of decoders based on linear programming, referred to as linear programming decoders. Considering linear programming relaxation of the Hamming norm decoders and solving them by using standard linear programming procedures, the linear programming decoders have polynomial complexities and are much more affordable. Both classes of decoders recover data from fewer received packets and hence achieve higher throughputs and shorter delays than the full rank decoder. Since these decoders could produce erroneous outputs, within each class several different decoding strategy have been proposed for different tradeoffs between delays/throughput and data accuracy, and they include the traditional decoder of network coding as a special case.

In the literature, there are two different approaches to deal with the synergy of network coding and compressive sensing, and they also aim for different applications. One approach was proposed in [3], statistical property of data blocks are taken advantage of to alleviate the "all-or-nothing" drawback of network coding in distributed storage systems. In this approach, random linear network coding is carried out over some finite fields, and the data are represented by bits. The other approach [4], [5] aims to take advantage of the statistical correlation of data generated by distributed sensor networks. A salient feature of this approach is that in theory data are real values and linear combinations are now performed over the real (or complex) field. The rationale for this is that the real representation of data is a more natural one for sensor networks [4], [5]. In practice, data are represented in a finite precision system. It has been shown that information loss due to finite precision grows with the network size [6].

Our work is quite different from both existing approaches. Above all, our reformulation of the decoding problem in network coding is novel, and this reformulation was not considered in the open literature to the best of our knowledge. Furthermore, the approach in [3] focuses on the application of random linear network coding in distributed storage systems.

In contrast, we consider linear network coding in general, and our work applies to a wide variety of applications. Also, network coding is carried out over the real (or complex) field in the approach in [4], [5], whereas in our work network coding remains over some finite fields. Thus, our scheme does not suffer the information loss due to finite precision as the approach in [4], [5].

## II. Rank Deficient Decoding

### A. System Model

In this work, we make several assumptions about the underlying network coding. First, we treat all packets as $N$-dimensional row vectors over some finite field GF($q$), where $q$ is a prime power. For simplicity, as most network coding schemes in practice, we assume that GF($q$) is a finite field of characteristic two, because information (in bits) can be easily mapped to finite field symbols. Second, we focus on linear network coding (LNC) [7] only, which was shown to be optimal in most cases. Thirdly, we assume that the network is error-free, and error control (see, for example, [8]–[11]) is not embedded in network coding.

Suppose a source node of a unicast or multicast injects a collection of $n$ data packets (or vectors over GF($q$)), $X_0, X_1, \cdots, X_{n-1}$, into the network. At any sink node, $m$ packets (or vectors over GF($q$)), $Y_0, Y_1, \cdots, Y_{m-1}$, are received, where $Y_i = \sum_{j=0}^{n-1} a_{i,j} X_j$ for $i = 0, 1, \cdots, m-1$ and $a_{i,j} \in$ GF($q$). In other words, each received packet is a linear combination of the injected packets. Since the sink node can locally generate more linear combinations of $Y_0, Y_1, \cdots, Y_{m-1}$, it is assumed that $Y_0, Y_1, \cdots, Y_{m-1}$ are linearly independent, which implies that $m \leq n$. That is, the $m \times n$ matrix $A = [a_{i,j}]$, often called the global coding kernel matrix, has a rank of $m$.

### B. Full Rank Decoding

Let us further denote the matrices $\left[X_0^T\, X_1^T\, \cdots\, X_{n-1}^T\right]^T$ and $\left[Y_0^T\, Y_1^T\, \cdots\, Y_{m-1}^T\right]^T$ as $X$ and $Y$, respectively, and they are related by $Y = AX$. The sink node can recover the transmitted data packets by reversing the encoding of the data packets by the network. This is easily achievable when $m = n$, as the sink node can recover the data packets by computing $X = A^{-1}Y$. Thus, the decoding in network coding starts only after the sink node has received an enough number of linearly independent combinations of the transmitted data packets. That is, either all data packets are recovered simultaneously, or none is recovered. This is often referred to as the "all-or-nothing" property by network coding. Note that this is different from the coupon collector problem suffered by communication networks without network coding. Nevertheless, the required number of linearly independent packets received by the sink node leads to longer delays and lower throughputs, which may be undesirable for some applications. Furthermore, in this setting, the "all-or-nothing" property also holds on the bit level. That is, all bits of all packets are equal in the sense that, either all of them are recovered simultaneous or none of them can be recovered.

### C. Rank Deficient Decoding

We can formulate the data recovery problem at the sink node in a different way. Let us consider coordinate $l$ of $Y_i$, and we have $Y_{i,l} = \sum_{j=0}^{n-1} a_{i,j} X_{j,l}$ for $i = 0, 1, \cdots, m-1$ and $l = 0, 1, \cdots, N-1$. Let us denote the column vectors $(Y_{0,l}\, Y_{1,l}\, \cdots\, Y_{m-1,l})^T$ and $(X_{0,l}\, X_{1,l}\, \cdots\, X_{n-1,l})^T$ as $Z_l$ and $W_l$, respectively. Clearly, we have $Z_l = AW_l$ for $l = 0, 1, \cdots, N-1$. The sink node can recover the data packets if it can obtain $W_l$ from

$$Z_l = AW_l \text{ for } l = 0, 1, \cdots, N-1. \tag{1}$$

Eq. (1) shows that the data recovery problem at the sink node can be viewed as $N$ parallel decoding problems in Eq. (1), and each corresponds to one coordinate in the packet (or vector). When these $N$ parallel decoding problems are solved at the same time, it is essentially equivalent to the traditional decoding problem of network coding.

This reformulated problem is related to two well known decoding problems. First, if we treat the $m \times n$ matrix $A$ as a parity check matrix for a linear block code of length $n$ and dimension $n - m$, the decoding problem in Eq. (1) is closely related to a syndrome decoding problem. That is, the sink node needs to recover $W_l$ based on the syndrome $Z_l$. Second, if we treat $W_l$ as a data vector and $A$ a measurement matrix, this is analogous to the decoding problem in compressive sensing.

### D. Decoding Strategy

Once a full rank of received packets are available, the full-rank decoder recovers all data packets correctly. In contrast, the proposed rank deficient decoders may produce wrong decisions. Analogous to classical error control coding, the preference between decoding failures and decoding errors varies from one application to another. For instance, for military applications with stringent delay constraints, partially correct data packets may be more desirable than decoding failures. For other applications such as cloud storage, data integrity may be a top priority than delays, especially packet retransmission is possible. Hence, it is necessary to consider a wide range of decoding strategy so as to offer different tradeoffs between delays/throughputs and accuracy.

Two extreme strategy are natural and straightforward. One extreme, called the error-free decoder, is similar to the full-rank decoder in the sense that it decodes only if decoding success is guaranteed. This can be implemented based on Lemma 1: decode only if $W_l$ in Eq. (1) satisfies $w_H(W_l) < \frac{d_H(A)}{2}$ for all $l$'s. The other extreme, referred to as the best-effort decoder, always tries to decode with available received packets. The error-free and best-effort decoders represent the most conservative and the most aggressive strategy.

We also devise a family of decoding strategy that fill the gap between these two extremes. These decoding strategy are based on one observation about error control codes. For an $(n, k)$ perfect code over GF(2), we have $\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k}$, where $t = \left\lfloor \frac{d_H(A)-1}{2} \right\rfloor$. In other words, all the coset leaders are unique and have Hamming weight up to $t$. However, since

most codes are not perfect and some allowance needs to be made. Hence, we devise a greedy-$l$ decoding strategy: decodes only if $\sum_{i=0}^{cw-l}\binom{n}{i} = 2^{n-k}$. The parameter $l$ represents how aggressive the decoder is: for the same code defined by $A$, the greater $l$ is, the more aggressive the decoder is. In fact, one can use different $l$ values to approach the two extremes, the best-effort and error-free decoders.

### *E. Hamming Norm Decoders*

Let us further consider the problem in Eq. (1). Since the data recovery problem at any sink node is equivalent to a collection of parallel problems in Eq. (1), we focus on one such problem. In other words, we try to solve $Z = AW$ for $W$, where $Z$ and $W$ are $m$- and $n$-dimensional column vectors, respectively, and $A$ remains an $m \times n$ matrix with full rank. Without loss of generality, we assume that $m < n$.

For a linear block code of length $n$ and dimension $n - m$ with a parity check matrix $A$, $Z = AW$ can be viewed as a syndrome of the received vector $W$. It is well known that for a linear block code, the syndromes have a one-to-one correspondence with its cosets, each of which is of size $q^{n-m}$. In other words, all vectors in a coset lead to the same syndrome. Thus, solving $Z = AW$ for $W$ is equivalent to finding a vector within a coset.

If no side information is available, we can make a decision within the coset by taking advantage of some inherent properties of the data vector. In this work, we proceed by relying on the sparsity of the data vector, which is well justified in many applications. That is, the proposed rank deficient decoders produce the vector with the smallest Hamming weight in the coset. Hence, we refer to them as Hamming norm decoders.

As is common in compressive sensing literature, we consider two possible scenarios for sparsity. First, when $W$ is sparse, we use a vector with the smallest Hamming weight in the coset corresponding to $Z$ as the estimate of $W$. Second, suppose that $\Phi W$ is sparse for a known nonsingular $n \times n$ matrix $\Phi$. Since $Z = AW = A\Phi^{-1}\Phi W$, we can treat $Z$ as a syndrome for the linear block code defined by $A\Phi^{-1}$. Thus, in this scenario, we first select a vector with the smallest Hamming weight in the coset corresponding to $Z$, and then produces an estimate of $W$ by multiplying the selected vector with $\Phi^{-1}$. In both scenarios, the key step is to select a vector with the smallest Hamming weight in the coset corresponding to the given syndrome. Thus, we assume $W$ is sparse without loss of generality.

In coding theory terminology, a vector with the smallest Hamming weight among a coset is called a leader of the coset. Note that some coset leaders may not be unique, when more than one vector in the coset has the smallest Hamming weight. In this case, either the coset leader is selected among these vectors at random or a list of all potential leaders.

We remark that this problem is closely related to but different from the syndrome decoding problem in classic coding theory. In our decoding, a vector or a list of vectors with the smallest Hamming weight in the coset corresponding to the given syndrome is considered as the estimate of the data vector. In the syndrome decoding problem, a coset leader is often considered as an estimate of the error vector. However, the key step in both problems is to select a vector or a list of vectors with the smallest Hamming weight in the coset corresponding to the given syndrome. For this reason, we refer to our decoding problem the modified syndrome decoding problem.

Thus, we have the following sufficient condition for successful decoding:

**Lemma 1.** *The minimum Hamming distance of the linear block code defined by A, denoted by $d_H(A)$, satisfies $d_H(A) \leq m+1$. When the Hamming weight of $W$, denoted by $w_H(W)$, is less than half of the minimum Hamming distance of the linear block code defined by A, that is $w_H(W) < \frac{d_H(A)}{2}$, $W$ can be recovered by syndrome decoding.*

*Proof:* The first part is due to the Singleton bound on the minimum Hamming distance of linear block codes. The second part holds because it is well known that a coset leader with Hamming weight less than $\frac{d_H(A)}{2}$ is unique. ■

When $W$ is not a unique coset leader, there are two possibilities. First, when the Hamming weight of $W$ is minimal in its coset, either $W$ has a probability to be selected when coset leaders are chosen at random or $W$ is one of the possible vectors produced by the decoder, depending on whether the decoder needs to generate only one vector or a list of vectors. Second, when the Hamming weight of $W$ is not minimal, a wrong vector will be produced by the modified syndrome decoder.

### *F. Linear Programming Decoders*

Since both the computational complexity and the memory requirement of the Hamming norm decoders grow exponentially with the size of $A$, we also adopt a linear programming (LP) approach. Since $A$ is not necessarily sparse, we formulate the problem based on that for binary linear block code with high-dense polytopes in [12].

An $m \times n$ parity-check matrix $A$ can be represented by a Tanner graph $G$, a bipartite graph with a set of variable nodes $\mathcal{I} = \{1, 2, \ldots, n\}$ and a set of check nodes $\mathcal{J} = \{1, 2, \ldots, m\}$. A node $i \in \mathcal{I}$ is adjacent to a node $j \in \mathcal{J}$ if the element $A_{i,j}$ is nonzero. $N(j)$ is the set of variable nodes that are adjacent to a check node $j$, and $N(i)$ is the set of check nodes adjacent to a variable node $i$.

Let $f_1, f_2, \ldots, f_n$ be the variables representing the code bits of $\mathbf{w}$, and $\mathbf{s} = (s_1, s_2, \ldots, s_m)^T$ be the syndrome received. For each check node $j \in \mathcal{J}$, let $T_j^E = \{0, 2, 4, \ldots, 2\lfloor|N(j)|/2\rfloor\}$ for $s_j = 0$, and $T_j^O = \{1, 3, 5, \ldots, 2\lfloor(|N(j)| - 1)/2\rfloor + 1\}$ for $s_j = 1$. Then for each $j \in \mathcal{J}$ and $k \in T_j^E$ $(T_j^O)$, define a variable $\alpha_{j,k}$. For all $j \in \mathcal{J}$, $k \in T_j^E$ $(T_j^O)$ and $i \in N(j)$, define $z_{i,j,k}$. Then the linear programming formulation for the syndrome decoding is to minimize $\sum_{i=1}^{n} f_i$, subject to the following constraints:

$$\forall i \in \mathcal{I}, j \in N(i), \qquad f_i = \sum_{k \in T_j} z_{i,j,k}$$

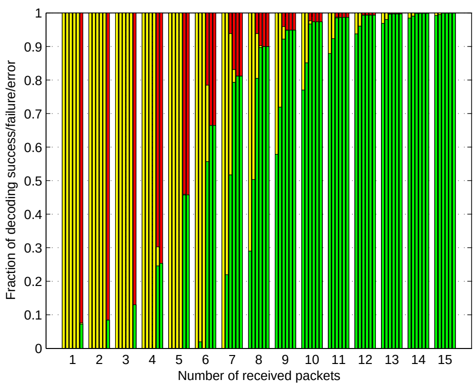


Fig. 1. Standard array vs LNC decoding: packets level

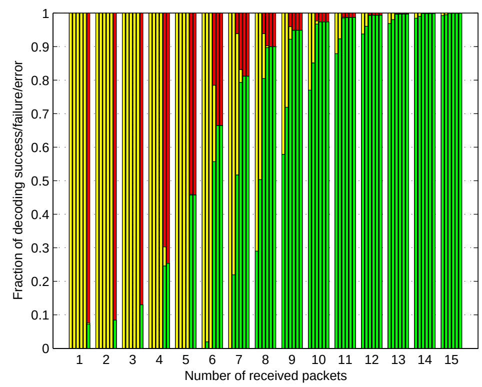


Fig. 2. Standard array vs LNC decoding: bits level

$$\forall j \in \mathcal{J}, \qquad \sum_{k \in T_j} \alpha_{j,k} = 1$$

$$\forall j \in \mathcal{J}, k \in T_j, \qquad \sum_{i \in N(j)} z_{i,j,k} = k\alpha_{j,k}$$

$$\forall i \in \mathcal{I}, \qquad 0 \leq f_i \leq 1$$

$$\forall j \in \mathcal{J}, k \in T_j, \qquad 0 \leq \alpha_{j,k} \leq 1$$

$$\forall i \in \mathcal{I}, j \in N(i), k \in T_j, \qquad 0 \leq z_{i,j,k} \leq \alpha_{j,k}.$$

The above constraints are similar to those in [12], except that $T_j = T_j^E$ in the previous constraints if $s_j = 0$, and $T_j = T_j^O$ if $s_j = 1$. In addition, the following constraint is added to narrow down the optimal solutions:

$$\sum_{i=1}^{n} f_i \geq cw$$

.

Linear programming may produce non-integral results, in which case two approaches are considered. The first type is simply to round off the real numbers into integers, which are compared with the original data to count decoding error or success rate, and we mark this approach LP I. The other one is to declare declare decoding failure of the entire generation, as the decoding is performed column wisely, and each packet in the same generation will be affected. Both LP I and LP II are applicable to all greedy as well as the EF and BE strategies.

## III. Simulation Results

In our simulations, $n = 8$ transmitted packets of length $N = 8$ are generated such that the transmission matrix has a constant column weight of $cw = 2$. Note that such small parameters are chose so that the complexities of Hamming norm decoders are manageable. The matrix $A$ is generated randomly, with each element being 0 or 1 with equal probability. The number of received packets $m$ varies from 1 to 15, while the packets may not be linearly independent. Simulation results are obtained based on 100,000 generations of packets injected into the network.

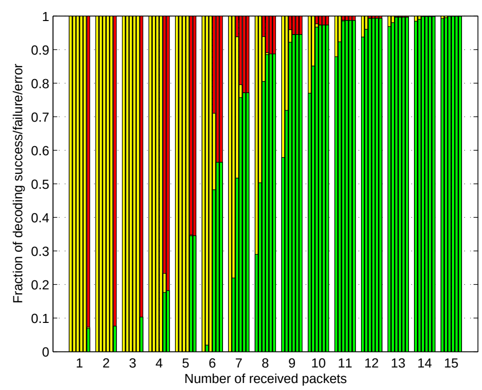


Fig. 3. Linear programming vs LNC decoding: packets level

Fig. 1 and Fig. 2 show the packet level and bit level performance of different decoding algorithms, where the syndrome decoding adopts the standard array decoding algorithm. Both the PSR and BSR approach 1 following increased number of received packets for the syndrome decoding and the LNC decoding. But LNC performs no decoding when the number of received packets $m$ is smaller than the number of transmitted packets $n$. Further, when $m \geq n$, the syndrome decoding algorithm achieves much better results than the traditional LNC for both the packet and bit levels performance.

Simulation results obtained from the linear programming decoding algorithm for the syndrome decoding are shown in Fig. 3 and Fig. 4. As expected, the linear programming approach performs slightly worse compared to the standard array decoding algorithm. However, the performance difference vanishes when the number of received packets is larger enough.

To measure the throughputs of these decoders, the average numbers of packets required to reach a packet success rate (PSR) of 1 or a bit success rate (BSR) of 0.95 are compared in Table I. The BE strategy requires approximately 2 less packets than the FR strategy to ensure all the packets are decoded correctly, for both the SA and the LP decoding algorithms.

TABLE I
AVERAGE NUMBER OF PACKETS FOR 100% PSR AND 95% BSR

| Strategy | FR | EF | greedy-(-1) | | | greedy-0 | | | greedy-1 | | | BE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SA | LP I | LP II | SA | LP I | LP II | SA | LP I | LP II | SA | LP I | LP II |
| 100% PSR | 9.60 | 8.84 | 8.12 | 8.12 | 8.12 | 7.57 | 7.78 | 7.77 | 7.44 | 7.73 | 7.72 | 7.44 | 7.73 | 7.72 |
| 95% BSR | 9.60 | 8.84 | 8.05 | 8.05 | 8.05 | 7.40 | 7.50 | 7.63 | 7.17 | 7.37 | 7.54 | 7.17 | 7.37 | 7.54 |

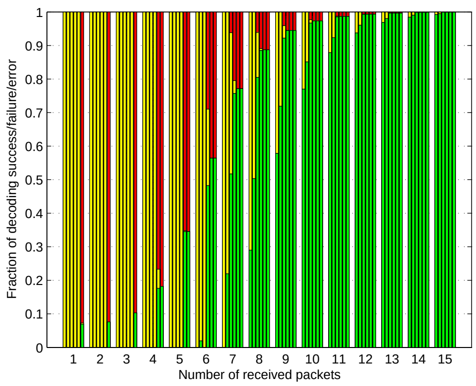


Fig. 4. Linear programming vs LNC decoding: bits level

Note that the numbers are the same as the 100% PSR case for both the FR and the EF strategies. However, for the other four strategies, the number of packets required for a 95% BSR is smaller than that for a 100% PSR. For the last three strategies, the LP I algorithm requires about 0.2 to 0.3 more packets compared to the SA decoding algorithm in order to decode all the packets correctly. The corresponding increase for a 95% BSR is about 0.1 to 0.2. For the LP II algorithm, both the PSR and the BSR decrease as the failure rates are slightly higher. As a result, 0.1 to 0.2 more packets are required to to reach a 95% BSR for the last three strategies, while the average numbers for a a 100% PSR remain the same for both algorithms.

## IV. CONCLUSIONS

We have proposed a variety of rank deficient decoders of linear network coding. Compared with the full rank decoder universally used in linear network coding, our proposed decoders require fewer received packets to decoder and hence achieve higher throughputs and shorter delays.

## REFERENCES


[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Info. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] T. Ho, M. Médard, R. Kötter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Info. Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.

[3] H. Chen, "Distributed file sharing: Network coding meets compressed sensing," *Proceedings of First International Conference on Communications and Networking in China (ChinaCom'06)*, pp. 1–5, October 2006.

[4] S. Katti, S. Shintre, S. Jaggi, D. Katabi, and M. Médard, "Real network coding," *Proceedings of Forty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, pp. 389–395, September 2007.

[5] N. Nguyen, D. Jones, and S. Krishnamurthy, "Netcompress: Coupling network coding and compressed sensing for efficient data communication in wireless sensor networks," *Proceedings of 2010 IEEE Workshop on Signal Processing Systems (SiPS 2010)*, pp. 356–361, October 2010.

[6] S. Shintre, S. Katti, S. Jaggi, B. K. Dey, D. Katabi, and M. Médard, "'Real' and 'complex' network codes: Promises and challenges," *Fourth Workshop on Network Coding, Theory and Applications (NetCod 2008)*, pp. 1–6, January 2008.

[7] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Info. Theory*, vol. 49, no. 2, pp. 371–381, February 2003.

[8] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. IEEE Information Theory Workshop*, Bangalore, India, October 2002, pp. 20–25.

[9] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. Info. Theory*, vol. 54, no. 1, pp. 209–218, January 2008.

[10] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Info. Theory*, vol. 54, no. 8, pp. 3579–3591, August 2008.

[11] H. Mahdavifar and A. Vardy, "Algebraic list-decoding on the operator channel," in *Proc. IEEE Int. Symp. Info. Theory*, Austin, USA, June 2010, pp. 1193–1197.

[12] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Info. Theory*, vol. 51, no. 3, pp. 954–972, March 2005.